

GEOMETRY-ADAPTIVE BLOCK PARTITIONING FOR INTRA PREDICTION IN IMAGE & VIDEO CODING

Congxia Dai^{†‡}, Óscar Divorrra Escoda[†], Peng Yin[†], Xin Li[‡] and Cristina Gomila[†]

[†]Thomson Corporate Research, Princeton, New Jersey 08540, USA

[‡]West Virginia University, Department of CS&EE, Morgantown, West Virginia 26506, USA

ABSTRACT

Many modern video coding strategies, such as the H.264/AVC standard, use quadtree-based partition structures for coding intra macroblocks. Such a structure allows the coding algorithm to adapt to the complicated and non-stationary nature of natural images. Despite the adaptation flexibility of quadtree partitions, recent studies have shown that these are not efficient enough (in terms of rate-distortion performance) when images can be locally modeled as 2D piecewise-smooth signals. These observations motivate us to investigate the use of geometry based block partitioning for modeling intra data in video coding. In particular, in this paper, we study in detail the use of geometry-adaptive intra models, where wedgelet like discontinuities are used in order to define separate coding regions where different statistical/waveform modeling tools can be used. In order to implement this idea, we extend the existing H.264/AVC intra coding scheme by introducing two additional geometric modes: INTRA16X16GEO, and INTRA8X8GEO. Experimental results show that significantly improved R-D performance is achieved.

Index Terms— Video Coding, Intra Prediction, H.264/AVC, Geometric Block Partitioning, Piecewise-Smooth Signals

1. INTRODUCTION

Intra prediction is an effective method to reduce spatial redundancy in intra coded video data. In state of the art video coding standard H.264/AVC [1], intra prediction modes (e.g. INTRA16X16, INTRA8X8(FRext), and INTRA4X4) are defined based upon a hierarchical quadtree-partition of intra data. For each partition, a set of prediction schemes using decoded neighboring information, has been carefully designed. This coding strategy has several advantages over previous video coding standards (e.g. H.263). First, the quadtree partitioning structure adapts automatically to the non-stationary nature of natural images: bigger blocks are often used to represent smooth regions, while smaller blocks tend to aggregate around edges or textured regions. Second, directional prediction schemes exploit some geometric redundancy helping to represent edge information and/or oriented textures by extrapolating samples from previously decoded neighboring pixels [2].

Despite the aforementioned advantages, H.264/AVC intra prediction strategy can still be further improved. For instance, if we consider a piecewise smooth image model (as illustrated in Fig. 1), where two different smooth regions, with different smoothness properties, are separated by an edge, small blocks will tend to accumulate around the boundary due to the difficulty of predicting both regions with a single model. In near-edge areas, tree-based partition leads to separately code different blocks with similar data with unnecessary overhead. Indeed, H.264/AVC standard does not take into account large scale geometry in images for efficient intra or inter coding. Be-

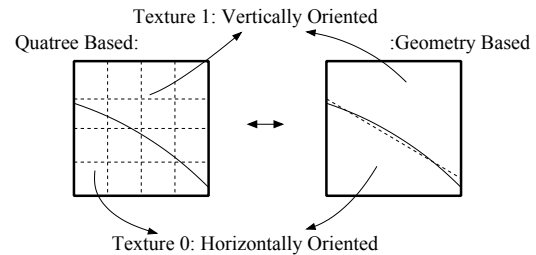


Fig. 1. Quadtree (left) vs Geometry (right) partitioning of 2D piecewise smooth signals.

sides that, another limitation in H.264/AVC is the limited usage of decoded neighboring pixels for prediction. Indeed, the farther a pixel is from the block boundary, the less accurate the prediction is, leading to losses in coding efficiency. This is the case shown in Fig. 5, where visually obvious staircase artifacts appear around edges.

Based on the observed limitations, in this paper, we present in detail a more flexible way of representing video intra data. The present work builds on concepts recently developed on the field of geometry-based representations for R-D efficient image coding. Indeed, theoretical studies have underlined the importance of modeling images as 2D piecewise-smooth signals for such a purpose [3]. For this reason, in this work, we investigate the use of geometry-adaptive block partitioning for intra coding purposes. In addition, and in the case of inter coding, recent works on the use of edge-adapted block partitions have shown promising results [4, 5, 6]. In [6], we proposed a general framework for geometry-adaptive block partitioning for both inter and intra video coding. In that work, the studied intra prediction scheme is in an early stage of research: only 16x16 block-size modes were considered for geometric partition, and a fixed, non-adaptive, prediction scheme was used to code partitions. In this paper, we focus, exclusively on intra coding. We explore and present in detail a complete geometry-adaptive intra partition structure. This includes 16x16 and 8x8 blocksize modes for geometric partition as well as the use of flexible prediction/modeling schemes such as directional prediction modes adaptively selected for every geometrically partitioned block region. The results obtained in here, show a significant improvement with regards to our previous work [6], and reflect as well a considerable performance improvement with respect to H.264/AVC based intra coding.

The remaining of the paper is organized as follows: Sec. 2 briefly reviews the motivations for using geometry-adaptive block partitioning for video intra coding. In Sec. 3, we describe in detail two geometric modes used to extend the H.264/AVC intra coding scheme: INTRA16X16GEO and INTRA8X8GEO. Experimental results are presented and discussed in Sec. 4. Finally, conclusions are drawn in Sec. 5.

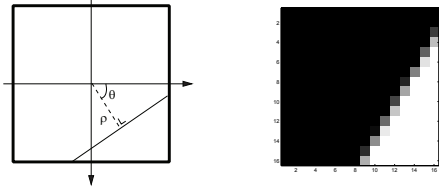


Fig. 2. Left: Line partition of a block based on geometric parameters θ and ρ . Right: Example of wedge-like partition with $\theta = \pi/6$ and $\rho = 4$. White color indicates one of the partitions, black marks the complementary partition. Gray intermediate values show “partial-surface” pixels.

2. ENHANCING INTRA CODING EFFICIENCY VIA GEOMETRY-ADAPTIVE BLOCK PARTITIONING

Natural images are often seen as bi-dimensional data with piecewise-smooth characteristics [3]. Considering such a signal model, it has been shown that quadtree-based structures are R-D suboptimal for coding purposes. Indeed, in some cases, more proper models for object boundaries and region edges should be considered for improved coding algorithms. In particular, geometry-adaptive block partitioning has been proved to be more R-D efficient for coding such kind of signals [3, 7]. In previous work [6], we discuss how video coding can benefit from geometry-adaptive block partitioning for intra and inter data coding. Indeed, it is well known that for piecewise models of an image such as the *Horizon* model, the asymptotic R-D behavior of the classic quadtree coding structure would be: $D_{Tree}(R) \sim \frac{\log R}{R}$. However, when wedgelet-like geometry partitioning is used within tree blocks, the R-D asymptotic behavior will approach that of the oracle based coding method: $D_{Wedge-Tree}(R) \sim \frac{\log R}{R^2}$, leading to a faster decay of distortion with rate.

Geometry-adaptive block partitioning allows to adaptively select different models for each partition depending on the signal while considering the geometric structure of object boundaries. This may help to mitigate some of the limitations of intra prediction schemes used in H.264/AVC, e.g. reducing the prediction error around edge regions as shown in Fig. 5. Unlike quadtree partitions, where a uniform prediction scheme is used for each block, the geometry based structure provides the flexibility to represent a block with two different models split by an edge, which can describe object boundaries more accurately and with lower overhead.

3. GEOMETRY-ADAPTIVE BLOCK PARTITIONING FOR INTRA PREDICTION

In this section, we present two new geometry-based intra prediction modes: INTRA16X16GEO, and INTRA8X8GEO. These modes, defined on 16x16 macroblocks and 8x8 blocks respectively, are introduced in H.264/AVC in order to evaluate the concepts discussed above. For this purpose, we review, first, the partition definition (as introduced in [6]), and then we discuss the different coding modes proposed in this paper.

3.1. Defining Geometry Based Block Partitions

Geometric partitions within blocks are defined by the implicit parametric model of a line: $f(x, y) = x \cos \theta + y \sin \theta - \rho$. The partitioning line, generated by the zero level line of $f(x, y)$ is determined by its angle θ and distance ρ as shown in Fig. 2. Based on this parametric line model, partitions are defined such that each pixel (x, y) is classified as:

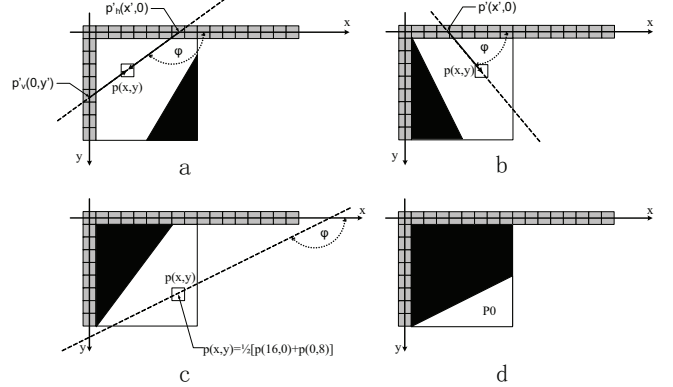


Fig. 3. The directional predicting(a,b,c) and DC modeling(d) schemes for partitioned regions (gray squares representing the neighboring predictors).

$$Partition(x, y) = \begin{cases} Partition\ 0 & \text{if } f(x, y) > 0 \\ Line\ Boundary & \text{if } f(x, y) = 0 \\ Partition\ 1 & \text{if } f(x, y) < 0 \end{cases} .$$

However, due to the discrete nature of digital images, the partitioning line may cross some pixels, and such pixels can not be fully classified in either partition (e.g. the “partial-surface” pixels in Fig. 2). In this case, we compute the prediction of these “partial-surface” pixels as a linear combination of their corresponding value if they were fully classified to each of the partitions.

For coding purposes, a dictionary of possible partitions is *a priori* defined such that $\rho : \rho \in [0, \sqrt{2}Blocksizel/2)$, $\rho \in \{0, \Delta\rho, 2 \cdot \Delta\rho, \dots\}$, and θ is such that $\theta \in [0, 2\pi)$ except for the case where $\rho = 0$, then $\theta \in [0, \pi)$. In here, $\Delta\rho$ and $\Delta\theta$ are the selected sampling steps for the radius and angle data respectively. Depending on the target bitrate, these can be modified in order to maximize R-D coding efficiency.

3.2. Predicting/Modeling schemes for Partitioned Regions

Given the geometric partitions defined above, for each partitioned region, a predicting/modeling scheme is selected using either the neighboring decoded information, or some model based on the statistics of the region.

Linear Directional Prediction: Considering a partitioned block of size $N \times N$ together with $3N + 1$ predictors (decoded neighboring pixels), the directional predicting scheme is defined such that every pixel $p(x, y)$ inside a partitioned region is predicted along the predicting direction φ from the predictors as shown in Fig. 3(a, b and c). Where $\varphi : \varphi \in [0, \pi)$, $\varphi \in \{0, \Delta\varphi, 2 \cdot \Delta\varphi, \dots\}$. More precisely, to predict pixel $p(x, y)$, the intersecting points of the line passing through (x, y) with orientation φ (the dashed line in Fig. 3) and the coordinate axes are determined. Pixel values at these intersecting points are linearly interpolated from the closest two predictors, and the predicted value of pixel $p(x, y)$ is linearly interpolated from the intersecting points (Fig. 3a). When there is only one intersecting point, the pixel value at this point is extrapolated from pixel $p(x, y)$ (Fig. 3b). If there is no such intersecting point within the range of the predictors, the prediction of pixel $p(x, y)$ is set to be the average of the two ending predictors (Fig. 3c).

Linear Directional Prediction is already a very useful tool for intra prediction in itself. This is why in the current H.264/AVC standard, directional prediction is considered at 4x4 and 8x8 block sizes. In the present work, in addition to these mode sizes, we also allow

the use of *Linear Directional Prediction*, as defined above, for 16x16 block sizes with no geometric partition. For comparison fairness, we keep the original H.264/AVC directional prediction algorithms for 4x4 and 8x8 block sizes.

DC Modeling: Sometimes, *Linear Directional Prediction* based on neighboring pixels is not able to provide accurate predictions for partitioned regions. For example, in Fig. 3d, pixels inside region P_0 might not be accurately predicted from neighboring pixels. In this case, a possibility is to model the region with a polynomial of certain order. In our current implementation, for the sake of simplicity, we choose a zero order polynomial to model that region, i.e. the DC value. Of course, more sophisticated models can be considered.

In our current settings, all geometric parameters are encoded as follows: The distance ρ was found close to an exponential distribution, hence, it is encoded by using Exp-Golomb codes. No predominant direction can be assumed *a priori* for angle θ , hence, this is encoded with fixed length codes. The prediction parameters φ and DC are differentially encoded using available neighboring information. φ is encoded differentially with respect to the partition angle θ , and DC mode value is predicted from neighboring decoded information when this is available. Exp-Golomb codes are used to code predicted φ and DC. Fixed length codes are used to code DC values when prediction is not possible. INTRA16X16GEO mode uses a bit to indicate whether the block is geometrically partitioned or not. Finally, each geometric partition spends an additional bit to indicate its prediction/modeling approach (directional or DC).

3.3. Selecting Optimal Geometric Parameters

At the encoding stage, for every macroblock, an optimal mode selection chooses among all possible intra prediction modes the one providing the lowest R-D cost: $J = D + \lambda R$. Where R is the total number of bits to encode the current macroblock, D is the distortion measure between the original macroblock and its decoded version (e.g. SSE or SAD), and λ is a Lagrangian multiplier. Hence, given a set of possible geometric block partitions: $\Gamma = \{P_1, P_2, \dots, P_N\}$, and a set of predicting/modeling parameters for each partitioned region: $\Omega = \{\varphi_1, \varphi_2, \dots, \varphi_M, DC\}$, the optimal geometric parameters: $(\gamma_{best}, \omega_{0_{best}}, \omega_{1_{best}})$ to represent current block should be selected such that $\forall \gamma \in \Gamma$, and $\forall \omega_0, \omega_1 \in \Omega$: $(\gamma_{best}, \omega_{0_{best}}, \omega_{1_{best}}) = \text{argmin}(J(\gamma, \omega_0, \omega_1))$.

Computational complexity is, certainly, an issue in geometry-adaptive image and video coding. According to the present problem formulation, for every possible partition, all directional/DC models need to be tested. In order to paliate as much as possible complexity increase, directional prediction pixels are computed just once for every block and stored into memory. Pre-computed directional predictions are, then, used over and over in the partition selection. Fast partition retrieval strategies are under research for further complexity reduction.

4. EXPERIMENTAL RESULTS

In this section, geometry extended H.264/AVC is compared to original H.264/AVC in terms of intra coding performance. In order to make a fair comparison, encoder settings of both intra coding schemes are exactly the same, with the exception of geometric extensions. Main used coding settings are: VLC coding, FRext extensions for intra coding, and deblocking filter is on. Concerning geometric extensions, the following parameter resolutions are used for all tested sequences (sequences are fully coded in intra mode): $\Delta\rho = 1$, $\Delta\theta_{16x16} = \frac{\pi}{16}$, $\Delta\theta_{8x8} = 2 \cdot \Delta\theta_{16x16}$ and $\Delta\varphi = \frac{\pi}{32}$. Throughout this section, average coding gains are given according

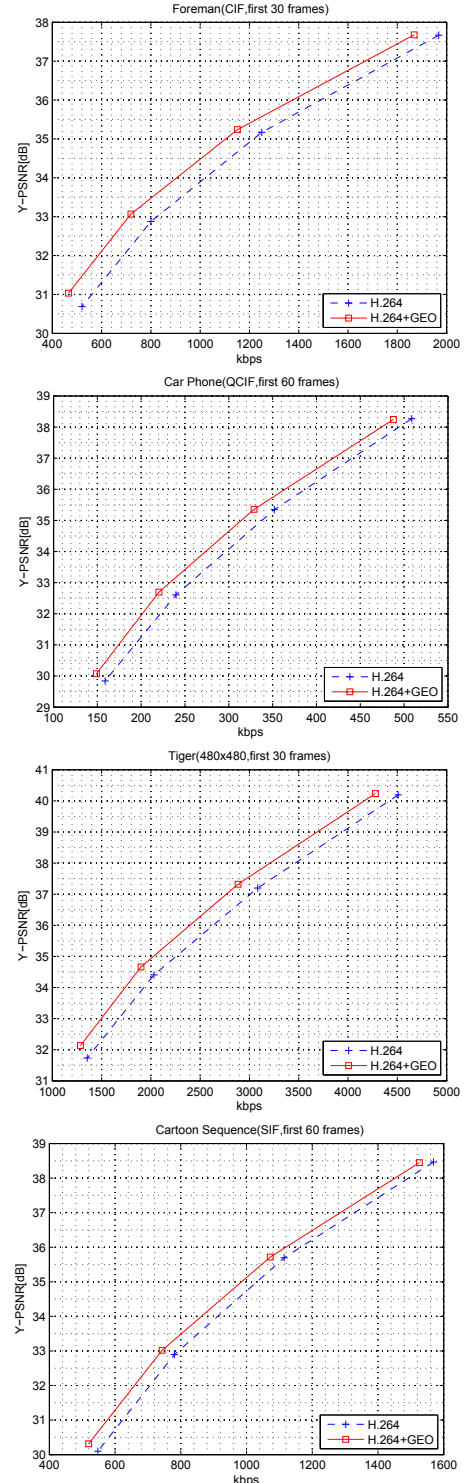


Fig. 4. R-D coding results. From top to bottom: Foreman (CIF), CarPhone (QCIF), Tiger (480x480) & Cartoon Sequence (SIF)

to the convention proposed in [8]. JSVM 6 reference code [9] has been used as H.264/AVC compliant codec for generating the results.

In Fig. 4, coding results of our geometric extended H.264/AVC for four test video sequences with different resolutions are presented. Compared to standard H.264/AVC intra coding, better R-D perfor-

Sequence	INTRA4x4	INTRA8x8	INTRA16x16	INTRA16x16_GEO	INTRA8x8_GEO
Foreman	37.37%	22.47%	10.86%	19.95%	9.34%
Tiger	21.44%	7.19%	38.22%	28.44%	4.69%
Cartoon	42.33%	8.92%	21.33%	22.00%	5.42%

Table 1. Mode Selection statistics for different sequences



Fig. 5. Experiment with the 15th frame of Foreman(CIF) at QP=28. Left: Intra geometric block partitioning edge map. Center: Intra Prediction (coded residual subtracted) by geometry extended H.264/AVC. Right: Intra Prediction (coded residual subtracted) by H.264/AVC.

mances can be observed. Average coding gains of 11.19%, 7.61%, 8.82%, and 5.23% are achieved for Foreman (CIF), Carphone (QCIF), Tiger (480x480), and a Cartoon sequence (SIF), respectively. Also, for Foreman and Tiger, if compared to the results reported in [6], an average extra bit savings of 2.5% and 3.17% are achieved, which are the joint contribution of INTRA8X8GEO mode and the more flexible directional prediction scheme introduced in this paper. As expected, we observe a relationship between QP and the required resolution for each geometric parameter in order to maximize the overall R-D performance. Indeed, low bit-rates (e.g. QP 40) require coarser geometric quantization than higher bit-rates. If this is considered in the coding scheme, a preliminary test showed that overall compression gains for Foreman raise to 12.15%. This suggests that geometry parameter quantization may be tuned depending on QP in order to achieve an overall optimal performance.

Fig. 5 shows the predicted versions of the 15th frame in Foreman sequence. By simple visual inspection, prediction improvements of geometry extended H.264/AVC can be easily observed. Edge regularity and region smoothness are significantly enhanced in Fig. 5 (middle) with respect to Fig. 5 (right). In terms of coding efficiency, this translates into a reduction of the residual energy, decreasing the amount of quantized non-zero transform coefficients. This demonstrates the edge preserving characteristics and better modeling capabilities of the proposed geometric intra prediction scheme. Fig. 5 (left) shows in detail which macroblocks are coded using the geometry-adaptive modes, as well as the encoded partition wedges. It can be observed that geometric partitions tend to aggregate around edges where H.264 standard intra prediction scheme produces poorer predictions. Also, some wedge partitions are used in non-edge areas in order to better model luminance changes and local gradients. In Table 1, percentages of blocks selecting each of the intra coding modes are listed for different sequences. It can be observed that quite a significant amount of blocks (around 30% for all three sequences) have chosen the new geometry-adaptive modes.

5. CONCLUSIONS

In this paper, a geometry-adaptive block partitioning scheme for intra video coding is presented. Intra coding performance of quadtree structured video codecs (e.g. H.264/AVC) is improved by adding

two geometric intra prediction modes: INTRA16X16GEO and INTRA8X8GEO. Currently, only linear block partitioning boundaries and very simple modeling tools for partitioned regions are used. In future, extensions of this work, we plan to investigate higher order partitioning boundaries together with more sophisticated region modeling tools (e.g. in order to model more textured areas, where present strategy is not yet successful) will be explored to further enhance intra coding performance. In addition, fast geometry-adaptive partition search strategies are required for reduced complexity encoding algorithms.

6. REFERENCES

- [1] ITU-T, *ITU-T Recommendation H.264*, Mar 2005.
- [2] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the h.264/avc video coding standard," *IEEE Trans on CSVT*, vol. 13, no. 7, pp. 560–576, 2003.
- [3] M. N. Do, P. L. Dragotti, R. Shukla, and M. Vetterli, "On the compression of two-dimensional piecewise smooth functions," in *ICIP*, Thessaloniki, Greece, Oct 2001.
- [4] S. Kondo and H. Sasai, "A motion compensation technique using sliced blocks in hybrid video coding," in *ICIP*, Genoa, Italy, Sep 2005.
- [5] E. M. Hung, R. L. De Queiroz, and D. Mukherjee, "On macroblock partition for motion compensation," in *ICIP*, Atlanta, USA, Oct 2006, IEEE.
- [6] Ò. Divorra Escoda, P. Yin, C. Dai, and X. Li, "Geometry-adaptive block partitioning for video coding," in *ICASSP*, Hawaii, USA, Apr 2007, IEEE.
- [7] V. Chandrasekaran, M. Wakin, D. Baron, and R. Baraniuk, "Surflats: A sparse representation for multidimensional functions containing smooth discontinuities," in *ISIT*, Chicago, USA, Jul 2004, IEEE.
- [8] G. Bjontegaard, "Calculation of average psnr differences between rd-curves," Tech. Rep., VCEG-M33, ITU-T Q.6/SG16, Apr 2001.
- [9] ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q. 6, *JVTM 6 Software*, Apr 2006.